PD&J Assessment 9

Prateek Bardhan
David Michaels
Jeremy Garrison

Intro to XGBoost

- Stands for Extreme Gradient Boosting
- Uses Gradient-Boosted Decision Trees (GBDT) to perform regression and classification
- One of the most popular machine learning algorithms
    - Gained popularity from winning Kaggle competitions
- Can be used with Python, C++, R, Julia, etc

Target Dataset

- XGBoost is a very versatile algorithm that can be used to perform linear regression, binary classification, multi classification, and more
  - Examples on the following slides
- XGBoost works best with large datasets

Linear Regression

Using Cars93 data and the housing dataset

```
bst = XGBRegressor(objective='reg:linear',random_state=42)
bst.fit(Xtr, ytr)
print("Training set accuracy score:",bst.score(Xtr,ytr))
print("Test set accuracy score:",bst.score(Xval,yval))
y_pred = bst.predict(Xval)
```

Binary Classification

Using the Pima Indians Diabetes dataset

```
xgb_model = xgb.XGBClassifier(objective="binary:logistic", random_state=42)
xgb_model.fit(X_train_whole, y_train_whole)
print("Training set accuracy score:",xgb_model.score(X_train_whole, y_train_whole))
print("Test set accuracy score:",xgb_model.score(X_test_whole, y_test_whole))
y_pred = xgb_model.predict_proba(X_test_whole)
```

Multi-class Classification

Using Kepler Exoplanet Search Results

```
xgb_model = xgb.XGBClassifier(booster = 'gbtree', eta = 0.25, max_depth = 5, objective="multi:softmax", random_state=42)
xgb_model.fit(X_train, y_train)
print("Training set accuracy score:",xgb_model.score(X_train, y_train))
print("Test set accuracy score:",xgb_model.score(X_test, y_test))
y_pred = xgb_model.predict(X_test)
```

Parameters

- The booster sets the type of learning, for example tree or linear
- The learning rate, denoted by eta, is the factor of shrinkage between each step
  - Between 0 and 1
  - Step shrinks by eta to avoid overfitting, default is 0.3
  - Eta too large makes computation faster with less steps, eta too small makes computation slower
- The max_depth parameter (default = 6) determines how deep the decision tree will go
- Different learning task parameters

Advantages

- One of the most popular Machine Learning Algorithms
  - Resources widely available online
- Several different options to handle regression, binary classification, and multi classification
- Efficient and quick with large datasets
- Easy to use with simple code to set up and run
- Several different hyperparameters to tune your model to avoid over/under fitting

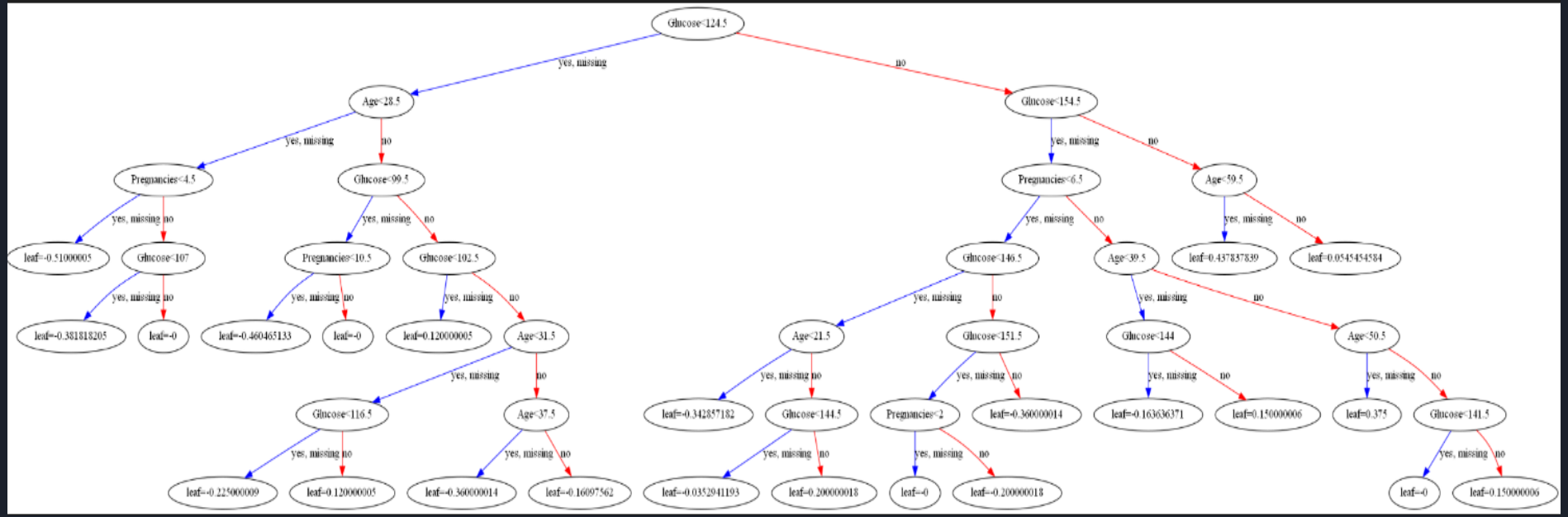
Disadvantages

- The numerous hyperparameters are a bit hard to understand
  - Lots of trial and error with the hyperparameters
- Needs the data in a DMatrix structure to run
  - Not difficult to convert a dataframe to a DMatrix, but an additional step nonetheless
- Struggles with a smaller dataset (i.e. Cars93)

Decision Tree - Diabetes

```
plot_tree(xgb_model)
fig = plt.gcf()
fig.set_size_inches(250, 100)
```

Pyth



Links

- https://www.kaggle.com/code/stuarthallows/using-xgboost-with-scikit-learn
- https://xgboost.readthedocs.io/en/latest/python/python_intro.html
- https://xgboost.readthedocs.io/en/stable/parameter.html
- https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-imbalanced-classification/#:~:text=An%20ROC%20curve%20(or%20receiver,True%20Positive%20Rate%20(y).
- https://machinelearningmastery.com/visualize-gradient-boosting-decision-trees-xgboost-python/
- https://www.youtube.com/watch?v=OQKQHNCVf5k
- https://www.youtube.com/watch?v=-D2Px4b0XQE
- https://www.youtube.com/watch?v=GrJP9FLV3FE

Links (continued)

- https://c3.ai/glossary/data-science/gradient-boosted-decision-trees-gbdt/#:~:text=Gradient%2Dboosted%20decision%20trees%20are%20a%20popular%20method%20for%20solving,to%20a%20sufficiently%20optimal%20solution.
- https://www.openintro.org/data/index.php?data=ames
- https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database
- https://www.kaggle.com/datasets/nasa/kepler-exoplanet-search-results

Questions?

Thank you!